CS1026 – Assignment #2
Volume Calculator

Due: Friday, May 28, 11:55pm on OWL
Weight: 12%

Learning Outcomes

By completing this assignment, you will gain skills relating to:

- using loops;
- using functions;
- using lists in Python;
- creating and using Python modules;
- testing programs and designing test cases; and
- following program specifications/requirements.

Task

In this assignment, you will write a **complete** program in Python that computes the volume for cylinders, prisms and spheres. Your program should make use of functions, loops and lists.

Your program will consist of two files: one is a module, `volumes.py`, which computes volumes, and the other is a main program, `main.py`, which uses the functions in module `volumes.py`. The main program, `main.py`, is to prompt the user for a type of object (e.g., a `‘cylinder’`) and validate that the object is one of the expected object types before computing the volume. In addition, your main program should keep track of each volume that is calculated. After the user chooses to `quit`, your program should display the volume for all the shapes entered in sorted order.

Functional Requirements/Specifications

1. Your main module, `main.py`, should handle the prompting and input for the different shapes and the output. Specifically, `main.py` should do the following:

a. Prompt the user for the shape they are interested in and check to make sure that their input is valid. Valid input options are as follows: `‘cylinder’` or `‘c’`; `‘prism’` or `‘p’`; `‘sphere’` or `‘s’`; and `‘quit’` or `‘q’`. **You should accept the input in any combination of upper- and lowercase letters (e.g., prism, PRisM, Prism, etc.)**. If the user enters an invalid option, your program should print a message and continue to prompt the user for a correct choice.

b. Continue to prompt the user for different shapes until the user enters `‘quit’` or `‘q’`; allow for any combination of uppercase and lowercase characters (e.g., `‘Quit’`, `‘QUit’`, `‘QUiT’`, etc.)

c. Prompt the user for the necessary dimensions for each of the respective shapes. **You may assume that the user enters positive floating-point values, so you DO NOT have to check the input of dimensions for valid values.**

d. Use the correct function in `volumes.py` to compute the volume of the specified shape.

e. Output a message with a computed volume.

f. Add the resulting shape and its volume to a list of volumes. Your program should have ONE list, and each item in the list should be a tuple of the form (shape_name, volume). The shape_name will be one of the following: `‘cylinder’`, `‘prism’`, or `‘sphere’` (all lowercase). You will have a single list of tuples that contains all the shapes entered and their computed volumes. For example, the list might look like `[(‘cylinder’, 1.00), (‘sphere’, 9.00)]`.

g. Once the user has entered `‘quit’` or `‘q’`, your program **should sort the list from lowest to highest.** To sort a list, `my_list`, of tuples, you can use the following version of the Python sort function:

```
my_list.sort(key = lambda my_list: my_list[1])
```

This tells Python to sort `my_list` using the element in position 1, which should be the volume in your tuple. Your program should then print the list of shapes and volumes **in sorted order**. The output should look like:

```
Output:  Volumes of shapes entered in sorted order:
prism 12.00
cylinder 37.70
sphere 229.85
prism 320.00
sphere 2065.24
```

Your output should consist of a header line exactly as above, and each shape and volume on separate lines. **All volumes should be printed with 2 decimals, and output should make use of the Python formatting operator %.**

h. If the user quits before entering any shapes, then the program should print out:

```
Output: No shapes entered.
```

2. Your module, `volumes.py`, should contain the functions for computing the volumes. **Each of your functions should calculate the volume of the shape and RETURN that value (i.e., it should not print a message).** Volumes for the different shapes are computed as follows:

Volume of a cylinder = $\pi r^2 h$

Volume of a rectangular prism = $l\ x\ w\ x\ h$

Volume of a sphere = $\frac{4}{3}\pi r^3$

You can search "volume of a" and the shape you want on Google, and you will get a calculator that shows the shape and provides a calculator. You can use this method to ensure your program is calculating the volumes correctly.

Non-Functional Requirements/Specifications

1. Include brief **comments** in your code identifying yourself (i.e., include your name in a comment at the beginning of your program), describing the program, and describing key portions of the code.

2. Assignments are to be done individually and **must be your own work**. Software may be used to detect academic dishonesty (cheating).

3. Use Python coding conventions and good programming techniques. Examples include:

    - meaningful variable names;
    - conventions for naming variables and constants (you can use underscores or camel case, but be consistent);
    - use of constants, where appropriate;
    - readability; and
    - indentation.

4. The name of the files you submit **MUST** be `volumes.py` and `main.py`. You will attach the files in OWL. **DO NOT** compress/zip them. **DO NOT** just enter code into the text submission area on OWL.

5. Make sure to use **Python 3.9** as the interpreter; failure to do so may result in the testing program failing.

Marking the Assignment

The TAs will be looking at the following things when grading your assignment:

1. Does the program behave according to the specifications found in the assignment document?

2. Does the program handle both valid and invalid input properly?

3. Is the output according to specifications (e.g., does it print in the format described in the specifications)? Is the calculated output correct?

4. Does the program follow the instructions for both input and output?

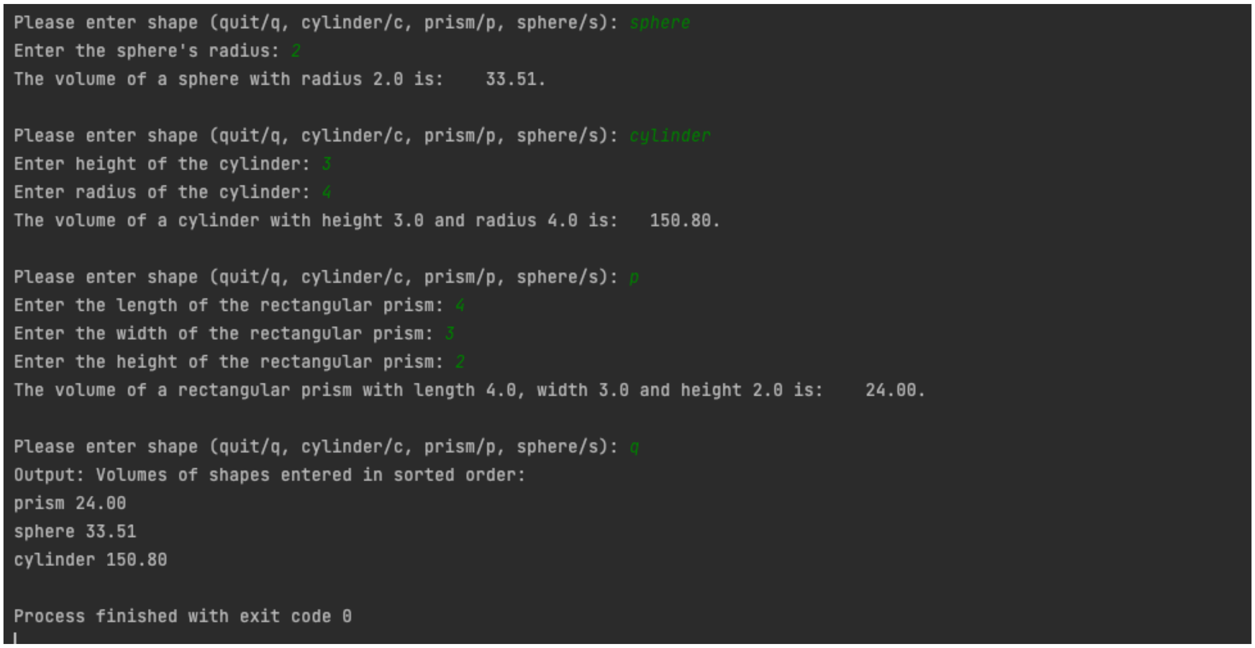
5. Does the submission meet the non-functional requirements (e.g., proper naming conventions, readability, proper file name, comments, etc.)?

The TAs will also be checking to ensure that things were not hardcoded and that your program actually uses the techniques learned in this course.

Examples of Output and Some Test Cases

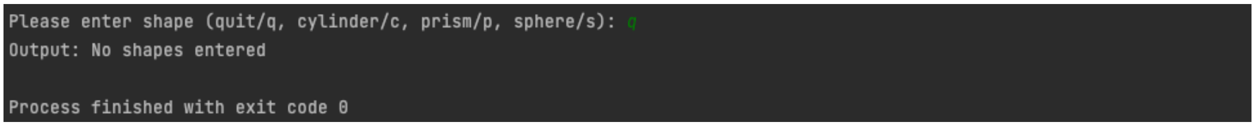
The following illustrate possible messages and output for your program and values for testing your program. These examples do not necessarily test all aspects of your program. It is your responsibility to design your own test cases and to test it thoroughly.

Example 1 – Regular Input



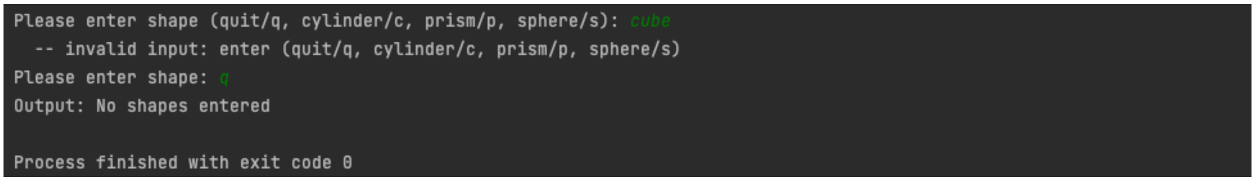
```
Please enter shape (quit/q, cylinder/c, prism/p, sphere/s): sphere
Enter the sphere's radius: 2
The volume of a sphere with radius 2.0 is:    33.51.

Please enter shape (quit/q, cylinder/c, prism/p, sphere/s): cylinder
Enter height of the cylinder: 3
Enter radius of the cylinder: 4
The volume of a cylinder with height 3.0 and radius 4.0 is:   150.80.

Please enter shape (quit/q, cylinder/c, prism/p, sphere/s): p
Enter the length of the rectangular prism: 4
Enter the width of the rectangular prism: 3
Enter the height of the rectangular prism: 2
The volume of a rectangular prism with length 4.0, width 3.0 and height 2.0 is:    24.00.

Please enter shape (quit/q, cylinder/c, prism/p, sphere/s): q
Output: Volumes of shapes entered in sorted order:
prism 24.00
sphere 33.51
cylinder 150.80

Process finished with exit code 0
```

Example 2 – No Input



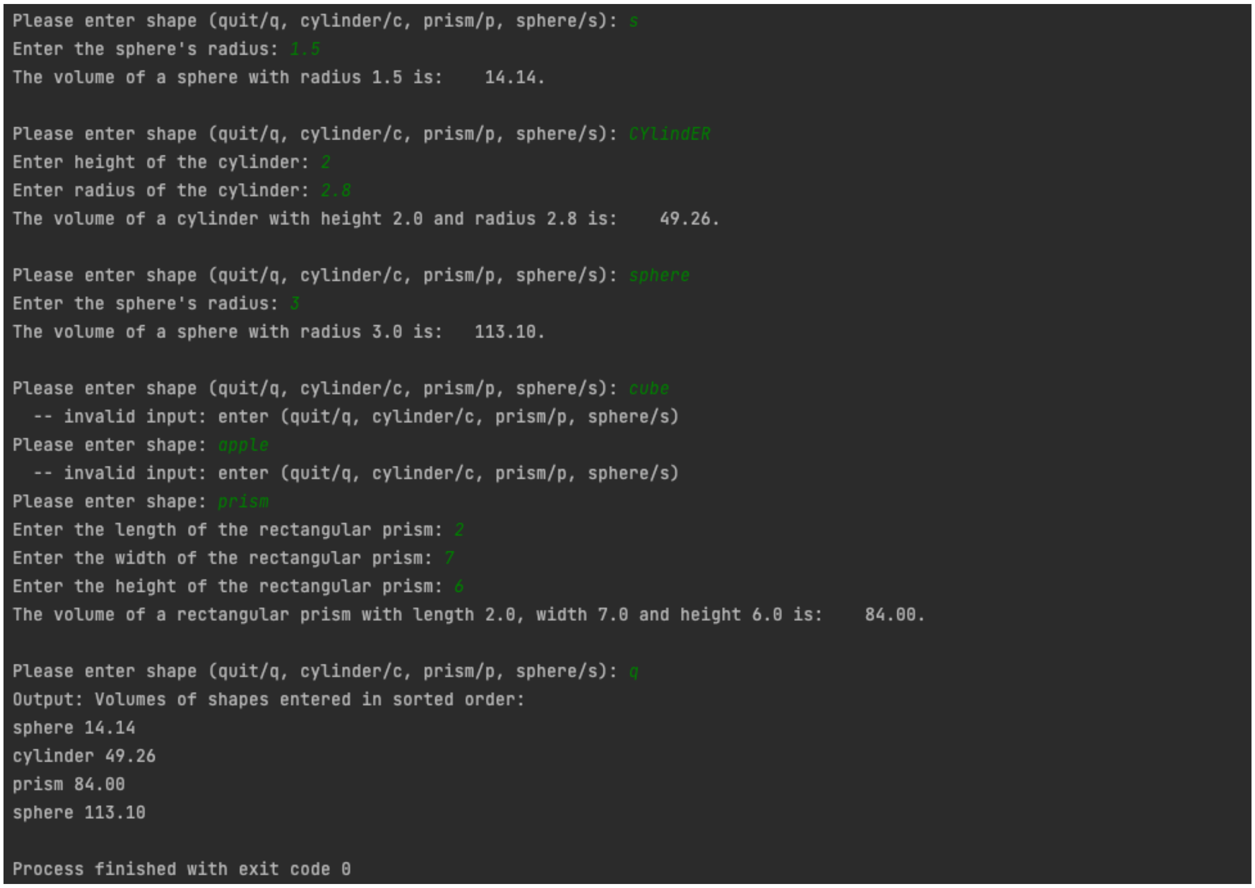
```
Please enter shape (quit/q, cylinder/c, prism/p, sphere/s): q
Output: No shapes entered

Process finished with exit code 0
```

Example 3 – Incorrect Input



```
Please enter shape (quit/q, cylinder/c, prism/p, sphere/s): cube
  -- invalid input: enter (quit/q, cylinder/c, prism/p, sphere/s)
Please enter shape: q
Output: No shapes entered

Process finished with exit code 0
```

Example 4 – A Variety of Input



```
Please enter shape (quit/q, cylinder/c, prism/p, sphere/s): s
Enter the sphere's radius: 1.5
The volume of a sphere with radius 1.5 is:    14.14.

Please enter shape (quit/q, cylinder/c, prism/p, sphere/s): CYlindER
Enter height of the cylinder: 2
Enter radius of the cylinder: 2.8
The volume of a cylinder with height 2.0 and radius 2.8 is:    49.26.

Please enter shape (quit/q, cylinder/c, prism/p, sphere/s): sphere
Enter the sphere's radius: 3
The volume of a sphere with radius 3.0 is:   113.10.

Please enter shape (quit/q, cylinder/c, prism/p, sphere/s): cube
  -- invalid input: enter (quit/q, cylinder/c, prism/p, sphere/s)
Please enter shape: apple
  -- invalid input: enter (quit/q, cylinder/c, prism/p, sphere/s)
Please enter shape: prism
Enter the length of the rectangular prism: 2
Enter the width of the rectangular prism: 7
Enter the height of the rectangular prism: 6
The volume of a rectangular prism with length 2.0, width 7.0 and height 6.0 is:    84.00.

Please enter shape (quit/q, cylinder/c, prism/p, sphere/s): q
Output: Volumes of shapes entered in sorted order:
sphere 14.14
cylinder 49.26
prism 84.00
sphere 113.10

Process finished with exit code 0
```